

PENGUNAAN METODE *FUZZY c-COVERING* UNTUK ANALISA MARKET BASKET PADA SUPERMARKET

Gregorius S. Budhi, Resmana Lim

Fakultas Teknologi Industri, Jurusan Teknik Informatika, Universitas Kristen Petra
e-mail: greg@petra.ac.id, resmana@petra.ac.id

Osmand Prayitno

Alumni Fakultas Teknologi Industri, Jurusan Teknik Informatika, Universitas Kristen Petra

ABSTRAK: Sebagai salah satu aplikasi data mining, market basket analysis umumnya dilakukan dengan memakai metode Apriori. Metode ini mencari asosiasi antar item dengan hanya menghitung berapa kali item-item tersebut muncul dalam keseluruhan transaksi tanpa memperhatikan *quantitas* item dalam transaksi. Oleh karena itu, peneliti mengusulkan metode *Fuzzy c-Covering*. Metode ini didasarkan pada persepsi bahwa semakin banyak item yang dibeli dalam suatu transaksi, maka hubungan antar item dalam transaksi tersebut semakin lemah. Untuk pengujian realisasi konsep tersebut, peneliti mengumpulkan data-transaksi penjualan dari sebuah swalayan selama satu bulan. Data yang didapat diubah ke bentuk yang dapat diolah oleh program. Kemudian sebuah algoritma dibuat untuk mengolah data tersebut dengan tujuan menghasilkan *association rules* dari item-item di dalam transaksi. Rule dapat ditampilkan dalam bentuk tabel ataupun grafik. Dengan menggunakan output dari program ini, yang berupa *association rule*, dapat diketahui tingkat asosiasi antar item yang berguna untuk membantu pihak pengambil keputusan dalam menentukan kebijakan pemasaran. Dari hasil pengujian, dapat disimpulkan bahwa semakin kecil minimum support dan confidence yang ditentukan, semakin banyak rule yang dapat dihasilkan dan waktu yang diperlukan semakin banyak. Selain itu, semakin tinggi jumlah kombinasi yang dicari, semakin sedikit waktu yang dibutuhkan.

Kata kunci: analisis *market basket*, *fuzzy c-covering*, program data mining, tingkat asosiasi antar item.

ABSTRACT: Data mining application with fuzzy c-covering method for market basket analysis on supermarket. As one of data mining applications, market basket analysis is generally performed using apriori method. However this method tends to search the association degree of items by only counting how many times the items appears on the overall transaction without considering the items quantity on each transaction. Therefore, we propose a new method which is based on perception that the more items bought in a transaction, the weaker the relation between items on that transaction. For the experimental purpose of the realization of the concept, we collect a month sales transaction data from a supermarket. The data are transformed to another form that can be used by the software. Then, an algorithm is created to process the data in order to generate association rule of items in transaction. The rules can be previewed in a table or graphic. Borland Cephli 7 and MS Access 2003 is used to build this experiment software. By using the output of this software, which are the association rules, association degree of items, can be obtained which is useful to help the decision maker to make market policies. Based on the testing result, it can concluded that the smaller minimum support and confidence, the more rule to be generated and the more processing time needed. Also, the higher the combination count to be searched, the less processing time needed.

Keywords: market basket analysis, association degree of items, data mining software, fuzzy c-covering.

PENDAHULUAN

Istilah *market basket* digunakan untuk menggambarkan kelompok *item* (terdiri dua atau lebih *item*) yang cenderung dibeli oleh seorang konsumen sewaktu berbelanja di perusahaan/swalayan dalam satu transaksi pembelian. Misalnya, kecenderungan konsumen untuk membeli *item* B jika ia membeli *item* A, atau sebaliknya. Jika hal itu bisa diketahui oleh swalayan, maka swalayan bisa meningkatkan penjualan *item* B dan *item* A dengan cara menempatkan keduanya saling berdekatan. Dengan cara itu, diharapkan konsumen akan teringat/tertarik untuk membeli *item* B jika ia akan membeli *item* A, atau

sebaliknya. Bisa juga kedua *item* itu diletakkan berjauhan dengan harapan agar dalam proses pencarian kedua *item* tersebut, konsumen tertarik untuk membeli *item* lain yang tidak direncanakan sebelumnya.

Selama ini, metode *data mining* yang dipakai untuk *market basket analysis* adalah metode apriori. Namun, metode yang dipakai ini cenderung memfokuskan pada hubungan antar *item* dalam keseluruhan transaksi saja tanpa memperhatikan hubungan tiap *item* dalam tiap transaksi. Misalnya jika mencari hubungan antara *Pepsodent* dan *Aqua*, metode ini hanya menghitung berapa kali keduanya itu muncul dalam keseluruhan transaksi tanpa memperhatikan jumlah *item* yang terdapat dalam

transaksi yang mengandung kedua *item* tersebut. Padahal, bisa saja walaupun pembelian keduanya banyak terjadi tetapi sebenarnya hal itu dipicu oleh *item* yang lain, misalnya pembelian *Pepsodent* dipicu oleh pembelian sikat gigi, sehingga hasil yang didapat tidaklah akurat. Oleh sebab itu sebuah metode *market basket analysis* yang baru diajukan. Dalam mencari hubungan antar *item*, metode ini berdasarkan pada persepsi bahwa semakin banyak *item* yang dibeli dalam suatu transaksi, maka hubungan antar *item* dalam transaksi itu semakin lemah.

TINJAUAN PUSTAKA

Market Basket Analysis

Market Basket Analysis, adalah salah satu cara yang digunakan untuk menganalisis data penjualan dari suatu perusahaan. Proses ini menganalisis *buying habits* konsumen dengan menemukan asosiasi antar *item-item* yang berbeda yang diletakkan konsumen dalam *shopping basket*. Hasil yang telah didapatkan ini nantinya dapat dimanfaatkan oleh perusahaan retail seperti toko atau swalayan untuk mengembangkan strategi pemasaran dengan melihat *item-item* mana saja yang sering dibeli secara bersamaan oleh konsumen.

Untuk beberapa kasus, pola dari *item-item* yang dibeli secara bersamaan oleh konsumen mudah untuk ditebak, misalnya susu dibeli bersamaan dengan roti. Namun, mungkin saja terdapat suatu pola pembelian *item* yang tidak pernah terpikirkan sebelumnya. Misalnya, pembelian minyak goreng dengan deterjen. Mungkin saja pola seperti ini tidak pernah terpikirkan sebelumnya karena minyak goreng dan deterjen tidak mempunyai hubungan sama sekali, baik sebagai barang pelengkap maupun barang pengganti. Hal ini mungkin tidak pernah terpikirkan sebelumnya sehingga tidak dapat diantisipasi jika terjadi sesuatu, seperti kekurangan stok deterjen misalnya. Inilah salah satu manfaat yang dapat diperoleh dari melakukan *market basket analysis*. Dengan melakukan proses ini secara otomatis, seorang manajer tidak perlu mengalami kesulitan untuk menemukan pola *item* apa saja yang mungkin dibeli secara bersamaan.

Association Rule

Association rule mining adalah suatu prosedur untuk mencari hubungan antar *item* dalam suatu *data set* yang ditentukan. (Han, Kamber, 2001). *Association rule* meliputi dua tahap (Ulmer, David, 2002):

1. Mencari kombinasi yang paling sering terjadi dari suatu *itemset*.
2. Mendefinisikan *Condition* dan *Result* (untuk *conditional association rule*).

Dalam menentukan suatu *association rule*, terdapat suatu *interestingness measure* (ukuran kepercayaan) yang didapatkan dari hasil pengolahan data dengan perhitungan tertentu. Umumnya ada dua ukuran, yaitu:

- **Support:** suatu ukuran yang menunjukkan seberapa besar tingkat dominasi suatu *item/itemset* dari keseluruhan transaksi. Ukuran ini menentukan apakah suatu *item/itemset* layak untuk dicari *confidence*-nya (misal, dari keseluruhan transaksi yang ada, seberapa besar tingkat dominasi yang menunjukkan bahwa *item* A dan B dibeli bersamaan).
- **Confidence:** suatu ukuran yang menunjukkan hubungan antar 2 *item* secara *conditional* (misal, seberapa sering *item* B dibeli jika orang membeli *item* A).

Kedua ukuran ini nantinya berguna dalam menentukan *interesting association rules*, yaitu untuk dibandingkan dengan batasan (*threshold*) yang ditentukan oleh *user*. Batasan tersebut umumnya terdiri dari *min_support* dan *min_confidence*.

Apriori Algorithm

"Apriori is an influential algorithm for mining frequent itemsets for Boolean association rules." (Han, Kamber, 2001). Apriori adalah suatu algoritma yang sudah sangat dikenal dalam melakukan pencarian *frequent Itemset* dengan *association rule*. Sesuai dengan namanya, algoritma ini menggunakan *knowledge* mengenai *frequent Itemset* yang telah diketahui sebelumnya, untuk memproses informasi selanjutnya. Algoritma inilah yang biasanya dipakai dalam proses *data mining* untuk *market basket analysis*. Algoritma apriori memakai pendekatan *iterative (level-wise search)*, dimana *k-itemset* dipakai untuk menyelidiki *(k+1)-itemset*. Langkah-langkah dari algoritma ini adalah sebagai berikut (Ulmer, David, 2002):

1. Set $k=1$ (menunjuk pada *itemset* ke-1).
2. Hitung semua *k-itemset* (*itemset* yang mempunyai k item).
3. Hitung *support* dari semua calon *itemset* – filter *itemset* tersebut berdasarkan perhitungan *minimum support*.
4. Gabungkan semua *k-sized itemset* untuk menghasilkan calon *itemset* $k+1$.
5. Set $k=k+1$.
6. Ulangi langkah 3-5 sampai tidak ada *itemset* yang lebih besar yang dapat dibentuk.
7. Buat *final set* dari *itemset* dengan menciptakan suatu *union* dari semua *k-itemset*.

Fuzzy c-Covering

Fuzzy c-Covering merupakan salah satu metode yang dipakai untuk mengklasifikasikan elemen-elemen dari suatu himpunan universal menjadi partisi-partisi berupa *fuzzy sets*. *Fuzzy c-Covering* sendiri merupakan generalisasi dari metode *fuzzy c-partition* yang telah dikenal sebelumnya. *Fuzzy c-partition* dapat didefinisikan sebagai berikut (Klir, Yuan, 2001):

Misalkan $I = \{i_1, i_2, \dots, i_n\}$ adalah *domain* dari data. *Fuzzy c-partition* dari I adalah *fuzzy subset* atau *fuzzy classes* dari T , ditunjukkan oleh $T = \{t_1, t_2, \dots, t_c\}$, yang memenuhi:

$$\sum_{m=1}^c \mu_{t_m}(i_k) = 1, \text{ untuk semua } k \in N_n. \quad (1)$$

$$0 < \sum_{k=1}^n \mu_{t_m}(i_k) < n, \text{ untuk semua } m \in N_c \quad (2)$$

di mana: c adalah *positive integer* ($0 < c \leq n$) dan $\mu_{t_m}(i_k) \in [0,1]$.

Berikut ini contoh untuk memperjelas rumus 1 dan 2.

Contoh 1: Dimisalkan ada suatu *fuzzy classes* sebagai berikut:

$$\mu_{t1} = \{1/i_1, 0.8/i_2, 0.2/i_3\},$$

$$\mu_{t2} = \{0.2/i_2, 0.8/i_3, 0.4/i_4\},$$

$$\mu_{t3} = \{0.6/i_4, 1/i_5, 1/i_6\}.$$

Dari contoh 1 dapat diketahui jumlah elemen yang terlibat ($n = 6$) dan jumlah partisinya ($c = 3$). *Fuzzy class* tersebut dapat dikatakan sebagai *fuzzy c-partition* dari suatu *domain* $I = \{i_1, i_2, i_3, i_4, i_5, i_6\}$ karena memenuhi rumus (1) dan (2). Di dalam *Fuzzy c-Covering*, rumus (1) dari teori *fuzzy c-partition* (Intan, Mukaidono, 2003) digeneralisasi menjadi:

$$\sum_{m=1}^c \mu_{t_m}(i_k) \geq 1, \text{ untuk semua } k \in N_n \quad (3)$$

Untuk memperjelasnya, diberikan contoh sebagai berikut:

Contoh 2: Dimisalkan ada suatu *fuzzy classes* sebagai berikut:

$$\mu_{t1} = \{1/i_1, 0.8/i_2, 0.4/i_3\},$$

$$\mu_{t2} = \{1/i_2, 0.8/i_3, 0.6/i_4, 0.2/i_6\},$$

$$\mu_{t3} = \{1/i_4, 1/i_5, 1/i_6\}.$$

Dari contoh 2 dapat diketahui $n = 6$ dan $c = 3$. *Fuzzy class* tersebut dapat dikatakan sebagai *Fuzzy C-Covering* dari suatu *domain* $I = \{i_1, i_2, i_3, i_4, i_5, i_6\}$ karena memenuhi rumus (2) dan (3).

Degree of Similarity in Fuzzy c-Covering

Pada contoh 2 dan 3 merepresentasikan *fuzzy classes* (T) sebagai *fuzzy set* terhadap *crisp element* dari data pada *domain* I . Sebaliknya, setiap *crisp*

element dari data pada I juga dapat direpresentasikan sebagai *fuzzy set* dari *fuzzy classes* (T) dengan rumus (Intan, Mukaidono, 2003):

$$\mu_{i1}(t_j) = \frac{\mu_{t_j}(i_1)}{\sum_{i \in I} \mu_{t_j}(i)} \quad (4)$$

Contoh 3: Berdasarkan contoh 2, elemen i_1 dari *fuzzy 3-partition* dapat direpresentasikan sebagai berikut:

$$\mu_{i1}(t_1) = \frac{\mu_{t1}(i_1)}{\mu_{t1}(i_1) + \mu_{t1}(i_2) + \mu_{t1}(i_3)} = \frac{1}{1 + 0.8 + 0.2} = 0.5$$

Bisa dikatakan bahwa $\mu_{i1}(t_1)$ adalah *true value* dari preposisi “if t_1 then i_1 ” atau *true value* dari i_1 given t_1 . Dengan cara yang sama, bisa didapatkan:

$$\mu_{i1} = \{0.5/t_1\},$$

$$\mu_{i2} = \{0.4/t_1, 0.14/t_2\},$$

$$\mu_{i3} = \{0.1/t_1, 0.57/t_2\},$$

$$\mu_{i4} = \{0.29/t_2, 0.23/t_3\},$$

$$\mu_{i5} = \{0.385/t_3\},$$

$$\mu_{i6} = \{0.385/t_3\}.$$

Seperti dapat dilihat, hasil perhitungan telah memenuhi rumus (1) dan (2). Cara yang sama juga bisa diterapkan pada setiap *crisp element* dari *Fuzzy C-Covering* dari contoh 3. Berikut adalah hasil perhitungannya:

Contoh 4:

$$\mu_{i1} = \{0.455/t_1\},$$

$$\mu_{i2} = \{0.364/t_1, 0.385/t_2\},$$

$$\mu_{i3} = \{0.182/t_1, 0.308/t_2\},$$

$$\mu_{i4} = \{0.231/t_2, 0.333/t_3\},$$

$$\mu_{i5} = \{0.333/t_3\},$$

$$\mu_{i6} = \{0.077/t_2, 0.333/t_3\}.$$

Seperti dapat dilihat, hasil perhitungan telah juga memenuhi rumus (1) dan (2).

Fuzzy Conditional Probability Relation

Umumnya, dalam mengukur derajat kesamaan antara dua *fuzzy set* digunakan *fuzzy conditional probability relation* dengan rumusan sebagai berikut: (Intan, Mukaidono, 2003)

$$R(X, Y) = P(X | Y) = \frac{|X \cap Y|}{Y} = \frac{\sum_{i \in I} \min\{\mu_X(i), \mu_Y(i)\}}{\sum_{i \in I} \mu_Y(i)} \quad (5)$$

di mana: μ_X dan μ_Y adalah fungsi anggota dari *domain* I untuk masing-masing X dan Y dari himpunan universal U .

Dalam prakteknya, *fuzzy conditional probability relation* dapat digunakan sebagai dasar untuk merepresentasikan *degree of similarity relationship* antara

dua *fuzzy set* dalam *universe* U . Dalam definisi *fuzzy conditional probability relation*, nilai probabilitas bisa diperkirakan berdasarkan hubungan semantik antar *fuzzy set* dengan menggunakan pandangan subjektif dari teori probabilitas. Berikut ini adalah contoh penggunaan rumus (5): Dari contoh 4. dapat dicari *similarity* $R(X,Y)$ dari tiap elemen dalam *Fuzzy c-Covering* sebagai berikut:

$$R(i_1, i_2) = \frac{\sum_{t \in I} \min\{\mu_{i_1}(t), \mu_{i_2}(t)\}}{\sum_{t \in I} \mu_{i_2}(t)} = \frac{0.364}{0.749} = 0.486$$

Dengan cara yang sama, dapat ditentukan juga hubungan tiap elemen lainnya. Hasilnya dapat dilihat pada Tabel 1.

Tabel 1. Similarity $R(X,Y)$ dari tiap elemen

X\Y	i_1	i_2	i_3	i_4	i_5	i_6
i_1	1	0.486	0.371	0	0	0
i_2	0.8	1	1	0.41	0	0.188
i_3	0.4	0.654	1	0.41	0	0.188
i_4	0	0.308	0.471	1	1	1
i_5	0	0	0	0.59	1	0.812
i_6	0	0.103	0.157	0.727	1	1

Tabel 1 menunjukkan suatu konsep yang menarik, yaitu bahwa suatu elemen memiliki hubungan (*similarity*) dengan yang elemen lainnya jika terlibat dalam kelompok (*class*) yang sama. Elemen-elemen tersebut akan memiliki hubungan yang lebih kuat jika sering terlibat dalam kelompok yang sama. Di sisi lain, dengan semakin meningkatnya jumlah elemen pada suatu kelompok akan mengurangi *degree of similarity* antar tiap elemen dalam kelompok tersebut. Berdasarkan konsep-konsep yang telah dijelaskan dapat dibuat suatu algoritma yang diterapkan dalam proses pembuatan aplikasi *data mining* untuk *market basket analysis*.

ALGORITMA UNTUK MARKET BASKET ANALYSIS BERDASARKAN FUZZY c-COVERING

Algoritma pengganti Algoritma *Apriori* berdasarkan *Fuzzy c-Covering*

Algoritma ini diajukan untuk memperbaiki kelemahan Algoritma *Apriori*. Dalam mencari hubungan antar *item*, metode ini berdasarkan pada persepsi bahwa semakin banyak *item* yang dibeli dalam satu transaksi, maka hubungan antar *item* yang terdapat dalam transaksi itu semakin lemah.

Berikut adalah langkah-langkah dalam algoritma yang dibuat berdasarkan metode baru tersebut:

Langkah 1: Menentukan *max_item_threshold* yang dibutuhkan. *Max_item_threshold* adalah suatu pembatas yang dipakai untuk menyaring transaksi berdasarkan jumlah *item* dalam transaksi tersebut. Hal ini didasarkan atas pemahaman bahwa semakin banyak *item* yang dibeli dalam suatu transaksi, hubungan antar *item* dalam transaksi tersebut semakin lemah.

Langkah 2: Mencari *record-record* dalam tabel transaksi yang memenuhi *max_item_threshold* dan menyimpannya ke dalam QT, dimana:

$$QT = \{t \mid |t| \leq \text{ith}, \text{ith} \in \text{positive integer}\}$$

di mana: QT (*Qualified Transaction*): himpunan transaksi yang memenuhi *max_item_threshold*; t : transaksi; $|t|$: jumlah *item* dalam suatu transaksi; *ith*: *max_item_threshold*.

Langkah 3: Set $k=1$ (k adalah variabel untuk menentukan jumlah kombinasi).

Langkah 4: Menentukan *min_support* ke- k sebagai *threshold* bagi kombinasi k -*item* terhadap tingkat dominasinya dari keseluruhan transaksi.

Langkah 5: Mencari *support* dari setiap kombinasi k -*item* yang memungkinkan yang ada di dalam transaksi tersebut dengan rumus:

$$\text{support}(u) = \frac{\sum_{t=1}^n \frac{1}{C_{|T_t|}^k s(u, T_t)}}{n} = \frac{\sum_{t=1}^n \frac{1}{\frac{|T_t|!}{k!(|T_t|-k)!} s(u, T_t)}}{n} \quad (6)$$

di mana: u : kombinasi k -*item* yang dicari *support*-nya. Jika I adalah *universal set of items*, maka $u \subseteq I$;

$|u| = k$: jumlah *item* dalam u ;

T_t : transaksi ke- t ($T_t \subseteq I$);

$|T_t|$: jumlah *item* dalam T_t .

$C_{|T_t|}^k$: kombinasi k -*item* terhadap $|T_t|$; n : jumlah *record/tuple* dalam QT;

$s(u, T_t) \in \{0,1\}$ adalah suatu *function*, dimana: jika $u \in T_t$, maka $s(u, T_t) = 1$, selain itu $s(u, T_t) = 0$.

Langkah 6: melakukan penyaringan terhadap kombinasi *item* yang ada di dalam transaksi tersebut yang tidak memenuhi: $\text{support}(u) \geq \text{min_support}$ ke- k .

Langkah 7: set $k=k+1$, dimana jika $k > \text{ith}$, maka ke langkah 9.

Langkah 8: mencari kombinasi k -*item* yang memungkinkan dari tiap kombinasi $(k-1)$ -*item* yang memenuhi *minimum support* yang telah ditentukan, dengan cara: untuk mendapatkan kombinasi k -*item*, u , harus ada semua kombinasi $(k-1)$ -*item*, u' , dimana

$u' \subset u$, misalnya untuk mendapatkan $u = \{I1, I2, I3, I4\}$, maka harus ada $u' = \{I1, I2, I3\}$, $\{I1, I2, I4\}$, $\{I1, I3, I4\}$ dan $\{I2, I3, I4\}$. Jika tidak ada lagi kombinasi k -item yang memungkinkan yang memenuhi $\min_support$ yang telah ditentukan maka ke langkah 9, selain itu ulangi langkah 4 - 7.

Langkah 9: Mendefinisikan tiap *item* yang telah didapat dari langkah-langkah di atas sebagai *fuzzy set* (disebut *item fuzzy set*) terhadap transaksi QT (menggunakan rumus (4) pada bab 2).

Langkah 10: Mencari *candidate rules* dengan cara menghitung *confidence* dari setiap kombinasi k -item yang memenuhi $\min_support$ ke- k ($k \geq 2$) dari *item fuzzy set* yang telah didapat pada langkah 9 dengan rumus:

$$R(X, Y) = \text{confidence}(Y \rightarrow X) = \frac{\sum_{t \in T} \inf_{i \in X \cup Y} (\mu_i(t))}{\sum_{t \in T} \inf_{i \in Y} (\mu_i(t))} \quad (7)$$

di mana: $X, Y \subseteq I$; T : himpunan dari kode-kode transaksi yang ada dalam QT; $\mu_i(t) \in [0, 1]$: fungsi anggota terhadap T .

Contoh untuk Memperjelas Cara Kerja Algoritma

Berikut ini diberikan contoh untuk memperjelas cara kerja algoritma, misalkan ada transaksi seperti pada Tabel 2.

Tabel 2. Tabel untuk Contoh 3.1

Kode transaksi	Item
100	I1, I2, I5
200	I2, I4
300	I2, I3
400	I1, I2, I4
500	I1, I3
600	I2, I3
700	I1, I3
800	I1, I2, I3, I5
900	I1, I2, I3
1000	I1, I2, I4, I6, I8

Misalkan $\max_item_threshold = 4$; set $k = 1$. Dari Tabel 3.1, yang memenuhi $\max_item_threshold$ hanya transaksi berkode 100 s/d 900. Maka dari Tabel 3.1, $QT = \{(I1, I2, I5), (I2, I4), (I2, I3), (I1, I2, I4), (I1, I3), (I2, I3), (I1, I3), (I1, I2, I3, I5), (I1, I2, I3)\}$ dan $T = \{100, 200, 300, 400, 500, 600, 700, 800, 900\}$. Kemudian tentukan $\min_support$ ke-1. Misalkan $\min_support_1 = 0.1 = 10\%$. Berdasarkan QT di atas, maka dapat ditentukan *item-item* yang dicari *support-*

nya, yaitu I1, I2, I3, I4 dan I5. Berikut ini adalah cara penghitungan *support* untuk tiap *item*:

$$I1 = \frac{\frac{1}{3} + 0 + 0 + \frac{1}{3} + \frac{1}{2} + 0 + \frac{1}{2} + \frac{1}{4} + \frac{1}{3}}{9} = \frac{1}{4} = 0.25;$$

$$I2 = \frac{\frac{1}{3} + \frac{1}{2} + \frac{1}{2} + \frac{1}{3} + 0 + \frac{1}{2} + 0 + \frac{1}{4} + \frac{1}{3}}{9} = \frac{11}{36} = 0.306;$$

$$I3 = \frac{0 + 0 + \frac{1}{2} + 0 + \frac{1}{2} + \frac{1}{2} + \frac{1}{4} + \frac{1}{3}}{9} = \frac{31}{108} = 0.287;$$

$$I4 = \frac{0 + \frac{1}{2} + 0 + \frac{1}{3} + 0 + 0 + 0 + 0 + 0}{9} = \frac{5}{54} = 0.093$$

(tidak memenuhi);

$$I5 = \frac{\frac{1}{3} + 0 + 0 + 0 + 0 + 0 + 0 + \frac{1}{4} + 0}{9} = \frac{7}{108} = 0.065$$

(tidak memenuhi).

Dari hasil perhitungan di atas, yang memenuhi $\min_support_1$ adalah I1, I2, dan I3. Setelah itu k diset menjadi $k=2$. Misalkan $\min_support_2 = 9\%$. Kemudian dicari kombinasi 2 -item dan *support* dari *item* yang tersisa:

$$\{I1, I2\} = \frac{\frac{1}{3} + 0 + 0 + \frac{1}{3} + 0 + 0 + 0 + \frac{1}{6} + \frac{1}{3}}{9} = \frac{7}{54} = 0.13;$$

$$\{I1, I3\} = \frac{0 + 0 + 0 + 0 + 1 + 0 + 1 + \frac{1}{6} + \frac{1}{3}}{9} = \frac{5}{18} = 0.28;$$

$$\{I2, I3\} = \frac{0 + 0 + 1 + 0 + 0 + 1 + 0 + \frac{1}{6} + \frac{1}{3}}{9} = \frac{5}{18} = 0.28.$$

Seperti yang dapat dilihat, semua kandidat *itemset* di atas memenuhi $\text{support}(u) \geq \min_support_2$, maka semua *itemset* tersebut dikombinasikan lagi menjadi kombinasi 3 -itemset. Set $k=3$ dan misalkan $\min_support_3 = 11\%$. *Support*-nya kemudian dicari sebagai berikut:

$$\{I1, I2, I3\} = \frac{0 + 0 + 0 + 0 + 0 + 0 + 0 + \frac{1}{4} + 1}{9} = \frac{5}{36} = 0.139$$

Karena sudah tidak ada kombinasi yang memungkinkan lagi untuk memenuhi $\min_support$ yang ditentukan, maka penghitungan selesai. Setelah itu tiap *item* yang telah didapatkan dari langkah-langkah di atas, yaitu I1, I2 dan I3, didefinisikan sebagai *fuzzy set* terhadap T . Untuk lebih jelasnya, akan diberikan cara mendefinisikan *item* I1 sebagai *fuzzy set* terhadap $T=100$ dan $T=400$ sebagai berikut:

$$\mu_{11}(100) = \frac{\mu_{100}(I_1)}{\mu_{100}(I_1) + \mu_{100}(I_2) + \mu_{100}(I_5)} = \frac{\frac{1}{3}}{\frac{1}{3} + \frac{1}{3} + \frac{1}{3}} = \frac{1}{3} = \frac{1}{3}$$

$$\mu_{11}(400) = \frac{\mu_{400}(I_1)}{\mu_{400}(I_1) + \mu_{400}(I_2) + \mu_{400}(I_4)} = \frac{\frac{1}{3}}{\frac{1}{3} + \frac{1}{3} + \frac{1}{3}} = \frac{1}{3} = \frac{1}{3}$$

Berdasarkan perhitungan tersebut, maka *item* I1, I2, I3 dan dapat didefinisikan sebagai berikut:

$$\mu_{11} = \{(1/3)/100, (1/3)/400, (1/2)/500, (1/2)/700, (1/4)/800, (1/3)/900\}$$

$$\mu_{12} = \{(1/3)/100, (1/2)/200, (1/2)/300, (1/3)/400, (1/2)/600, (1/4)/800, (1/3)/900\}$$

$$\mu_{13} = \{(1/2)/300, (1/2)/500, (1/2)/600, (1/2)/700, (1/4)/800, (1/3)/900\}$$

Dari *item-item* tersebut kemudian dicari *confidence* dari setiap kombinasi *k-item*, dimulai dari *k=2*.

Kombinasi 2-item:

$$confidence(I1 \rightarrow I2) = \frac{\frac{1}{3} + \frac{1}{3} + \frac{1}{4} + \frac{1}{3}}{\frac{9}{4}} = \frac{5}{9} = \frac{5}{9} = 0.56 = 56\%$$

(baca: *confidence* dari rule **if I1 then I2** sebesar 56%).

$$confidence(I2 \rightarrow I1) = \frac{\frac{1}{3} + \frac{1}{3} + \frac{1}{4} + \frac{1}{3}}{\frac{11}{4}} = \frac{5}{11} = 0.455 = 45.5\%$$

$$confidence(I1 \rightarrow I3) = \frac{\frac{1}{2} + \frac{1}{2} + \frac{1}{4} + \frac{1}{3}}{\frac{9}{4}} = \frac{19}{27} = 0.704 = 70.4\%$$

$$confidence(I3 \rightarrow I1) = \frac{\frac{1}{2} + \frac{1}{2} + \frac{1}{4} + \frac{1}{3}}{\frac{31}{12}} = \frac{19}{31} = 0.613 = 61.3\%$$

$$confidence(I2 \rightarrow I3) = \frac{\frac{1}{2} + \frac{1}{2} + \frac{1}{4} + \frac{1}{3}}{\frac{11}{4}} = \frac{19}{33} = 0.576 = 57.6\%$$

$$confidence(I3 \rightarrow I2) = \frac{\frac{1}{2} + \frac{1}{2} + \frac{1}{4} + \frac{1}{3}}{\frac{31}{12}} = \frac{19}{31} = 0.613 = 61.3\%$$

Kombinasi 3-item:

$$confidence(I1 \wedge I2 \rightarrow I3) = \frac{\frac{1}{4} + \frac{1}{3}}{\frac{1}{3} + \frac{1}{3} + \frac{1}{4} + \frac{1}{3}} = \frac{7}{15} = 0.47 = 47\%$$

(baca: *confidence* dari rule **if I1 and I2 then I3** sebesar 47%).

$$confidence(I1 \wedge I3 \rightarrow I2) = \frac{\frac{1}{4} + \frac{1}{3}}{\frac{1}{2} + \frac{1}{2} + \frac{1}{4} + \frac{1}{3}} = \frac{7}{19} = 0.368 = 36.8\%$$

$$confidence(I2 \wedge I3 \rightarrow I1) = \frac{\frac{1}{4} + \frac{1}{3}}{\frac{1}{2} + \frac{1}{2} + \frac{1}{4} + \frac{1}{3}} = \frac{7}{19} = 0.368 = 36.8\%$$

$$confidence(I1 \rightarrow I2 \wedge I3) = \frac{\frac{1}{4} + \frac{1}{3}}{\frac{1}{3} + \frac{1}{3} + \frac{1}{2} + \frac{1}{2} + \frac{1}{4} + \frac{1}{3}} = \frac{7}{27} = 0.259 = 25.9\%$$

$$confidence(I2 \rightarrow I1 \wedge I3) = \frac{\frac{1}{4} + \frac{1}{3}}{\frac{1}{3} + \frac{1}{2} + \frac{1}{2} + \frac{1}{3} + \frac{1}{2} + \frac{1}{4} + \frac{1}{3}} = \frac{7}{33} = 0.21 = 21\%$$

$$confidence(I3 \rightarrow I1 \wedge I2) = \frac{\frac{1}{4} + \frac{1}{3}}{\frac{1}{2} + \frac{1}{2} + \frac{1}{2} + \frac{1}{2} + \frac{1}{4} + \frac{1}{3}} = \frac{7}{31} = 0.226 = 22.6\%$$

Nilai *confidence* dipakai untuk menentukan *rule* mana saja yang merupakan *interesting rule*. Misalkan *user* menentukan *min_confidence*=60%, maka yang disebut sebagai *interesting rule* hanya *rule* yang nilai *confidence*-nya $\geq 60\%$, yaitu:

if I1 then I3 [*support* = 28%, *confidence* = 70.4%],

if I3 then I1 [*support* = 28%, *confidence* = 61.3%],

dan

if I3 then I2 [*support* = 28%, *confidence* = 61.3%].

DESAIN SOFTWARE

Software ini mempunyai tiga proses utama, yaitu:

1. **Proses 1 – Support Analysis:** Berfungsi untuk mencari kombinasi untuk setiap *item/itemset* yang didapatkan dari data penjualan yang di-input-kan *user* berupa tabel transaksi (Tabel 3.2) dan menentukan apakah *item/itemset* tersebut layak untuk dicari *rules*-nya. Hasil dari proses ini disimpan pada tabel *itemset*.
2. **Proses 2 – Rules Analysis:** Berfungsi untuk mencari dan menghasilkan *rules-rules* yang memungkinkan dari setiap *itemset* yang telah disimpan dalam tabel *itemset*. Setiap *rules* yang dihasilkan pada proses ini disimpan pada tabel *rules*.
3. **Proses 3 – View Analysis Result:** Berfungsi untuk merepresentasikan ke monitor *rules-rules* (yang telah didapatkan pada proses 2 dan disimpan pada tabel *rules*) yang sesuai dengan keinginan *user* untuk keperluan analisis. Hasil yang disajikan dapat berupa grafik maupun tabel.

PENGUJIAN

Hardware Yang Digunakan Untuk Pengujian

Pengujian program ini dilakukan pada komputer dengan spesifikasi sebagai berikut:

- Processor Intel Pentium III 450 MHz.
- Memory SDRAM 192 Mbyte.

- HardDisk 40 GByte.
- Operating System Windows 2000 Server.

Pengujian Software

Pengujian dilakukan sebanyak tiga kali menggunakan data transaksi selama satu minggu dan dua kali menggunakan data satu bulan, dengan beberapa *minimum support* yang berbeda. Diasumsikan *max_item_treshold* diabaikan dan *min_support* untuk tiap kombinasi bernilai sama.

Tabel 3. Hasil Pengujian Untuk Data Satu Minggu (1-7 Oktober 2002)

Jumlah	Support (%)	Support Analysis		Rule Analysis		
		ItemSet	Waktu	Confidence (%)	Jumlah Rules	Waktu
Transaksi 245 Barang 767	0.5	C-ItemSet	4 menit 36 detik	0	2	3 menit 51 detik
		1118		25	2	
		Q-ItemSet		50	2	
		28		75	2	
	0.4	C-ItemSet	22 menit 8 detik	0	16	14 menit 45 detik
		2686		25	15	
		Q-ItemSet		50	5	
		70		75	3	
	0.3	C-ItemSet	33 menit 55 detik	0	20	36 menit 42 detik
		3587		25	19	
		Q-ItemSet		50	7	
		85		75	3	

Tabel 4. Hasil Pengujian Untuk Data Satu Bulan (1-31 Oktober 2002)

Jumlah	Support (%)	Support Analysis		Rule Analysis		
		ItemSet	Waktu	Confidence (%)	Jumlah Rules	Waktu
Transaksi 1097 Barang 1059	0.3	C-ItemSet	19 menit 44 detik	0	4	7 menit 33 detik
		2334		25	4	
		Q-ItemSet		50	4	
		26		75	1	
	0.2	C-ItemSet	1 jam 54 menit 56 detik	0	30	1 jam 11 menit 46 detik
		8562		25	30	
		Q-ItemSet		50	23	
		53		75	8	
	0.1	C-ItemSet	10 jam 50 menit 50 detik	0	232	15 jam 44 menit 40 detik
		58734		25	207	
		Q-ItemSet		50	128	
		432		75	64	

Pada Tabel 3 dan 4 diatas dapat dilihat hasil pengujian yang telah dilakukan. Dari tabel tersebut dapat disimpulkan bahwa kecepatan program dalam melakukan penghitungan *support* dan pencarian *rules* tergantung pada total jumlah *item* yang terdapat di dalam transaksi serta *minimum support* yang di-inputkan oleh *user*. Semakin banyak *item* yang ada di dalam transaksi yang memenuhi *minimum support*,

semakin banyak *item* yang harus dikombinasikan untuk mendapatkan *candidate 2-itemset*.

Sebagai contoh, untuk 10 *item* dapat menghasilkan kombinasi *2-itemset* yang memungkinkan sebanyak $C_{10}^2 = \frac{10!}{2!(10-2)!} = 90$ kombinasi. Sedangkan

untuk 50 *item*, terdapat $C_{50}^2 = \frac{50!}{2!48!} = 1225$ kombinasi

2-itemset yang memungkinkan. Jumlah *item* yang dikombinasikan tergantung sekali pada jumlah *minimum support* yang ditentukan oleh *user*. Hal ini dikarenakan semakin besar *minimum support*-nya, semakin kecil kemungkinan suatu *item* untuk lolos dari seleksi sehingga semakin sedikit jumlah *item* yang harus dikombinasikan, dan begitu juga sebaliknya. Jika dimisalkan untuk menghasilkan 1 kombinasi dibutuhkan waktu 1 detik, maka untuk 10 *item* membutuhkan waktu 90 detik (1.5 menit), sedangkan untuk 50 kombinasi dibutuhkan 1225 detik atau sekitar 20.4 menit.

KESIMPULAN

Berdasarkan hasil pembahasan sebelumnya dapat disimpulkan bahwa:

- Metode baru yang diajukan ini, yaitu: Fuzzy c-Covering, dapat diterapkan pada pembuatan *software data mining* untuk *market basket analysis*.
- Dalam menentukan batasan (*threshold*) yang sesuai, tergantung pada tempat *software* ini digunakan dan periode dari data yang dianalisis.
- Dengan menggunakan *software* ini, setelah menyediakan data transaksi dan menjalankan *software*, manajer dapat mengetahui *item-item* apa saja yang saling berasosiasi pada sebuah pasar swalayan dan berapa besar tingkat asosiasi tersebut.
- Pada analisis terhadap sejumlah data, ditemukan bahwa semakin kecil *minimum support* dan *confidence* yang ditentukan, semakin banyak pula *rule* yang dapat dihasilkan oleh *software*. Namun, waktu yang diperlukan relatif lebih lama dibandingkan jika *minimum support* lebih besar.
- Dari pengujian ditemukan bahwa semakin tinggi jumlah kombinasi yang dicari, waktu yang dibutuhkan relatif lebih cepat. Hal ini terjadi karena semakin lama, *item/itemset* yang tersaring semakin banyak sehingga kombinasi yang perlu dicari semakin sedikit.
- Waktu yang diperlukan untuk pemrosesan tergantung pada spesifikasi komputer, jumlah transaksi dan jumlah *item* yang terlibat. Yang membuat pemrosesan data menjadi lama adalah pada waktu pencarian kombinasi tiap *itemset* dan penghitungan *confidence*.

DAFTAR PUSTAKA

1. Cyber Victory, 2003. *Combination Algorithm.*, <<http://www.cybvic.org.mm/resources/?pageid=27>>.
2. Dilly, Ruth, *Data Mining: An Introduction*, Parallel Computer Centre, Queens University Belfast, 2004
3. Han, Jiawei; Kamber, Micheline, *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2001.
4. Witten, Ian H.; Frank, Eibe, *Data Mining Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann Publishers, 2000.
5. Hand, David; Mannila, Heikki; Smyth, Padhraic, *Principles Of Data Mining*, The MIT Press, 2001.
6. Intan, Rolly; Mukaidono, Masao, *A Proposal of Fuzzy Thesaurus Generated by Fuzzy Covering*. IEEE. 2003, pp. 167-172.
7. Klir, George J.; Yuan, Bo, *Fuzzy Sets and Fuzzy Logic – Theory and Applications*. India: Prentice Hall, 2001.
8. Palace, Bill, *Data Mining: What Is Data Mining?*. 1996 <<http://www.anderson.ucla.edu/faculty/jason.frand/teacher/technologies/palace/damining.htm>>
9. Rusly, Hendra, *Pembuatan Software Data Mining untuk Membantu Pengambilan Keputusan dengan Menggunakan Metode Market Basket Analysis pada Perusahaan "X"*, Thesis, UK Petra, 2003.
10. Ulmer, David; "Mining an Online Auctions Data Warehouse." *The Mid-Atlantic Student Workshop on Programming Languages and Systems*. 19 April 2002. Pace University. <<http://csis.pace.edu/csis/masplas/p8.pdf>>.